

Symbolic Analysis of the Neuron Action Potential

Pei Ye*, Emilia Entcheva[†], Scott A. Smolka*, Radu Grosu*

*Computer Science Department
State University of New York at Stony Brook
Stony Brook, New York, 11794

[†]Department of BioMedical Engineering
State University of New York at Stony Brook
Stony Brook, New York, 11794

Abstract—We present a novel approach to investigating key behavioral properties of complex biological systems by first using automated techniques to learn a simplified *Linear Hybrid Automaton* model of the system under investigation, and then carrying out automatic reachability analysis on the resulting model. The specific biological system we consider is the *neuronal Action Potential* and the specific question of interest is *bifurcation*: the graded response of the neuron to stimulation of varying amplitude and duration. Reachability analysis in this case is performed using the d/dt analysis tool for hybrid systems. The results we so obtain reveal the precise conditions under which bifurcation manifests, when taking into consideration an infinite class of input stimuli of arbitrary shape, amplitude, and duration within given respective intervals.

To the best of our knowledge, this represents the first time that formal (reachability) analysis has been applied to a computational model of excitable cells. The obvious advantage of symbolic reachability analysis over simulation—perhaps the only available analysis method when complex systems of coupled ODEs are used to model excitable-cell behavior, as has traditionally been the case—is that through the so-called *reachable set computation*, the system’s reaction to an infinite set of possible inputs can be observed. Our results further demonstrate that Linear Hybrid Automata, as a formal language, is both expressive enough to capture interesting excitable-cell behavior, and abstract enough to render formal analysis possible.

I. INTRODUCTION

Hybrid automata [1], [2] are an increasingly popular modeling formalism for systems that exhibit both continuous and discrete behavior. Intuitively, hybrid automata (HA) are extended finite-state automata whose discrete states correspond to the various modes of continuous dynamics a system may exhibit, and whose transitions express the switching logic between these modes.

Excitable cells (ECs), which include neuronal, cardiac and skeletal-muscle cells, are a typical example of biological systems exhibiting hybrid behavior: transmembrane ion fluxes and voltages may vary continuously but the transition from the resting state to the excited state is generally considered an all-or-nothing discrete response. Traditionally, however, the preferred modeling approach for ECs uses large sets of coupled nonlinear differential equations. Although an invaluable asset for integrating genomics and proteomics data to reveal local interactions, such models are not typically amenable to

control-theoretic techniques developed for linear systems, and render large-scale simulation impractical.

At the cellular level, the electrical signal ECs amplify and propagate is a change in the potential across a cell’s membrane, caused by different ion currents flowing through the membrane’s channels. For each excitation event, the electrical signal in question is called an *action potential* (AP). For non-pacemaking ECs, an AP is an externally triggered all-or-nothing response to an external stimulus: if the stimulus is sufficiently strong, the cell fires an excitation AP; see the continuous curve in Figure 1(a). Otherwise, it returns to its resting potential; see the dashed curve in Figure 1(a). This so-called *bifurcation property* of ECs, which is typical for nonlinear systems, will be the main focus of our symbolic analysis.

Specifically, the bifurcation problem in this paper is the graded response of the neuron to stimulation of varying amplitude and duration. If the neuron under consideration can be stimulated directly, we can choose a simple input stimulus, such as a rectangular pulse of fixed height and width. The result for several such stimuli is shown in Figure 1(b) (data from running simulation of Hodgkin-Huxley model [3]). Unfortunately, the number of such rectangular pulses is potentially infinite. Moreover, if the cell interacts with neighboring cells, we can no longer assume this simple form of input. The lack of linearity also tells us that the output cannot be computed as the superposition of the outputs of the system for a given set of generators for the input. Hence, we are left with having to simulate the system for all possible input stimuli; even when we discretize the input with a sample-and-hold technique, the number of such stimuli is intractably large.

In this paper, we show that it is possible to apply automatic formal analysis techniques to this bifurcation analysis problem for ECs. Specifically, based on our learning technique of [4], we develop a simplified Linear Hybrid Automaton (LHA) model of the neuronal AP, which we submit for automatic symbolic reachability analysis, as implemented in the d/dt analysis tool for linear hybrid systems [5]. The results we so obtain reveal the precise conditions under which bifurcation manifests, when taking into consideration an infinite class of input stimuli of arbitrary shape, amplitude, and duration within

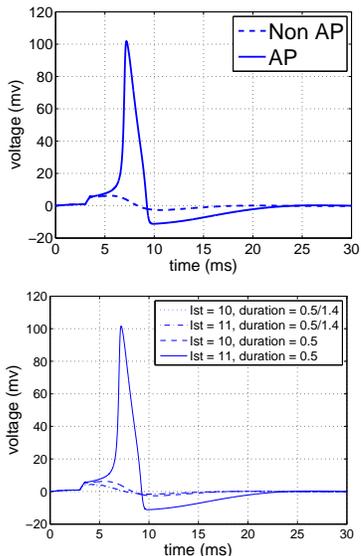


Fig. 1. (a) APs for the squid giant axon. (b) APs for four different input pulses.

given respective intervals.

To the best of our knowledge, this represents the first time that formal analysis has been applied to a computational model of excitable cells. The obvious advantage of the symbolic reachability analysis we conducted over simulation is that, through the so-called reachable set computation, the system's reaction to an infinite set of possible inputs can be observed. This form of analysis is simply not possible by simulation alone. Our results further demonstrate that LHA, as a formal language, is both expressive enough to capture interesting excitable-cell behavior, and simple enough to render formal analysis possible.

In related work, Dumas and Rondepierre [6], starting from a simplified Hodgkin-Huxley model [3], developed a piecewise linear Hybrid automaton for the neuron action potential, and studied bifurcation of the system through phase-space plotting. The major difference between this effort and the results presented herewith is that our analysis of the neuron action potential is based on the use of *automatic* formal-analysis techniques. In particular, we have shown how automatic symbolic reachability analysis, as implemented in the *d/dt* analysis tool, reveals the precise conditions under which bifurcation manifests in a simplified LHA model of the neuron action potential, when taking into consideration an infinite class of input stimuli of arbitrary shape and duration within given respective intervals.

II. HYBRID AUTOMATA

Definition [Hybrid automaton (HA)] An HA $\mathcal{A} = (X, G, \mathit{init}, \mathit{inv}, \mathit{flow}, \mathit{jump}, \mathit{event})$ over Σ is a 7-tuple consisting of [7]:

- A finite set X of real-valued variables x_1, \dots, x_n ; their dotted form $\dot{x}_i \in \dot{X}$ represents first derivatives and their primed form $x'_i \in X'$ represents values at the conclusion of discrete steps (jumps); n is called the *dimension* of \mathcal{A} .
- A finite control graph $G = (V, E)$, whose vertices in V are called *modes* and edges in E are called *switches*.

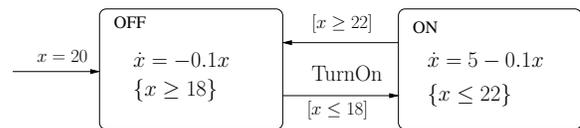


Fig. 2. A thermostat system modeled as an HA.

- For each mode $v \in V$, vertex-labeling functions init , inv and flow with domain V and range P , where P is the set of all logical predicates. Initial condition $\mathit{init}(v)$ and invariant $\mathit{inv}(v)$ are predicates with free variables from X . Flow $\mathit{flow}(v)$ is a predicate with free variables from $X \cup \dot{X}$ representing a set of ordinary (partial) differential (in)equalities.
- A finite set Σ of events which are essentially boolean variables controlled from outside the system, and an edge-labeling function $\mathit{event}: E \rightarrow \Sigma$ that assigns to each switch an event.
- Edge-labeling functions $\mathit{jump}: E \rightarrow (\mathit{Guard}, \mathit{Action})$ where Guard is a set of predicates with free variables from $X \cup \Sigma$ and Action is a set of assignments that update the variable in X' .

An HA \mathcal{A} spends time in its modes $v \in V$, where it updates its variables according to the flow predicate $\mathit{flow}(v)$. Jumps $\mathit{jump}(e)$ on switches $e = (v, w)$ are in contrast instantaneous, where v is the beginning mode and w is the end mode of the switch. A jump on e is taken whenever the jump's guard $\mathit{jump}(e).Guard$ is enabled for the current value of variables X , or the invariant $\mathit{inv}(v)$ of the current mode is violated.

For illustration purposes, consider HA \mathcal{A} of Figure 2, which models a simple thermostat system. This is a one-dimensional system over input alphabet $\Sigma = \{\text{TurnOn}\}$, with $X = \{x\}$ and x representing the current temperature; control graph $G = (\{\text{ON}, \text{OFF}\}, \{(\text{ON}, \text{OFF}), (\text{OFF}, \text{ON})\})$, with invariants $\mathit{inv}(\text{OFF}) = \{x \geq 18\}$ and $\mathit{inv}(\text{ON}) = \{x \leq 22\}$, flows $\mathit{flow}(\text{OFF}) = \{\dot{x} = -0.1x\}$ and $\mathit{flow}(\text{ON}) = \{\dot{x} = 5 - 0.1x\}$ and guards $\mathit{jump}((\text{OFF}, \text{ON})).Guard = \{x \leq 18\}$ and $\mathit{jump}((\text{ON}, \text{OFF})).Guard = \{x \geq 22\}$; and a single event given by $\mathit{event}((\text{OFF}, \text{ON})) = \text{TurnOn}$. Initially, \mathcal{A} is in mode OFF with x initialized to 20°C. When in this mode, the heater is off and the temperature drops until it falls to 18°C and the event TurnOn occurs. At this time, \mathcal{A} jumps to mode ON, in which the heater is on and the temperature rises until it reaches 22°C. At this point, \mathcal{A} jumps back to mode OFF.

Definition [Linear HA (LHA)] An HA is a LHA if it has the following properties:

- Linear flows. Every $\mathit{flow}(v)$ is a linear time-invariant differential equation of the form $\dot{X} = AX + BU$, where A is a constant $n \times n$ matrix, B is a constant scalar, and U is an input vector of dimension $n \times 1$ that ranges inside a convex polyhedron.
- Linear invariants and guards. The variable spaces that make $\mathit{inv}(v)$ and $\mathit{jump}(e).Guard$ true also form convex polyhedra.

The *reachability problem* for HA is stated as follows: *Will the system, subject to certain initial conditions, ever enter an “unsafe” state?* The problem is in general undecidable; however, under certain restrictions on the HA flows or jumps, like the ones we put on LHA, it becomes decidable [8]. A popular reachability analysis tool for LHA is d/dt [9].

The d/dt tool performs forward reachability analysis on LHA, presenting the final result in a graphical form. It uses convex polyhedral packages for this purpose, representing reachable sets of states as unions of convex polyhedra. In order for d/dt to perform reachable-set computations, one has to provide it with the following data:

- *Dimension*: The number of the variables in the input LHA.
- *Initloc*: The initial mode of the LHA.
- *Initset*: The initial-range polyhedron of the state variables.
- *Badset*: The unwanted polyhedron area, in the case of safety analysis.
- *Locations*: The modes of the LHA.
- *Matrix A* and *scale B*: Matrix and scale as in Definition II.
- *Inputset*: A polyhedron bounding the range of U for each mode.
- *Transitions*: The guard polyhedron and target mode for each switch.
- *Stayset*: The invariant polyhedron for each mode.
- *Limits*: The range polyhedron on which reachability analysis is performed.

III. BIFURCATION ANALYSIS METHOD

For the purpose of bifurcation analysis, we will restrict our attention to the derivation of a simple neuronal LHA. We will then input this LHA to d/dt in order to symbolically analyze its response to any possible input over an arbitrary time domain within a given time interval.

As in [10], [4], the automatic learning method we use for neuronal LHA proceeds in two stages:

- Identify the topology of the LHA, i.e. the design of its control graph.
- Identify for each control-graph mode, the dynamics of the LHA model.

The choice of modes is based on the observation that the AP for different cell types (neuron, cardiac myocyte, etc.) or different species (guinea pig, neonatal rat, etc.) may exhibit different waveforms, but they all possess the following two phases within a cycle: a *Rising* phase and a *Falling* phase; see Figure 1. For the purpose of mode identification, we also need to identify the time period during which the cell is subject to external stimulation. We shall refer to this mode as *Stimulated*, and allow the LHA model to accept input within this mode. This leads to the splitting of the rising phase into modes *Stimulated* and *Rising*.

We use a begin-stimulus event e_s and an end-stimulus event \bar{e}_s to effect this separation. When external stimulation begins, i.e., upon the occurrence of e_s , the LHA switches to mode *Stimulated*, in which the cell accumulates its membrane voltage by accepting an input current. If upon termination of

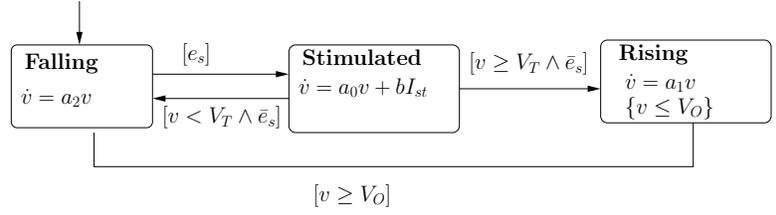


Fig. 3. Graphical representation of neuronal LHA.

the stimulation, i.e., upon the occurrence of the event \bar{e}_s , the magnitude of the received stimulation is sufficiently strong, the cell fires an AP by switching to mode *Rising*; otherwise it returns to mode *Falling*. In the former case, the LHA will switch from mode *Rising* to *Falling* when it passes the maximum voltage point.

Since the voltage is the only observable in which we are interested and since bifurcation, the phenomenon of interest, only occurs at the end of stimulation, we tolerate a larger error margin within mode *Falling*. As a consequence, our learning algorithm is able to derive an acceptable LHA with only one continuous state variable representing the AP. Certainly, this might be not the case for other observables. The LHA so derived is depicted in Figure 3.

The flow function is of the form $\dot{v} = a_0v + bI_{st}$ for mode *Stimulated* and $\dot{v} = a_i v$ ($i = 1, 2$) for the other two modes (these modes are therefore called refractory modes). The threshold values V_O and V_T are constants determined by analyzing the data produced by the HH model. The coefficients a_i for $i = 0, 1, 2$, and b are also automatically learned by analyzing the data produced by the HH model via numerical integration. Their precise values can be provided upon request.

IV. ANALYSIS RESULTS

Let us henceforth refer to the neuronal LHA of Figure 3 (where time is implicit) as LHA1. In this section, we first compare the simulation results for a single AP, resulting from a rectangular-pulse stimulus, using the LHA1 and Hodgkin-Huxley model (HH). We then present the results of our symbolic reachability analysis for LHA1 (with certain adaptation to fit the tool d/dt), where all input stimuli (of any form and duration within given intervals) are automatically (symbolically) taken into account by d/dt. Our results, given in terms of LHA1’s reachable state sets, clearly demonstrate the bifurcation property over this infinite input set.

a) Validation of the LHA model. Figure 4(a) compares the simulation of a single AP using the LHA1 and HH models. The initial condition for the voltage was the same in both models. The parameter values considered for the HH model were as follows: $\bar{g}_{Na} = 120\text{mOhms}^{-1}/\text{cm}^2$, $\bar{g}_K = 36\text{mOhms}^{-1}/\text{cm}^2$, $\bar{g}_L = 0.3\text{mOhms}^{-1}/\text{cm}^2$, $E_{Na} = 55.0\text{mv}$, $E_K = -72.0\text{mv}$, $E_L = -50.0\text{mv}$, $C = 1$. The stimulation current for both models was a rectangular pulse with an amplitude of $50\text{uA}/\text{cm}^2$ and a duration of 0.5ms . As can be observed, the AP is better matched during the rising phase than in the falling phase, as our focus with the neuronal LHA model is on whether or not an AP will be fired under

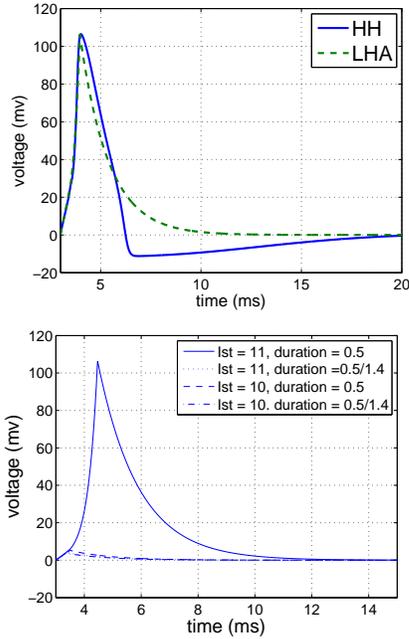


Fig. 4. (a) AP comparison: LHA1 vs. HH. (b) Bifurcation in LHA1

different input currents. The undershoot during the falling phase is missed in the LHA1 model, a compromise made in favor of the model's simplicity.

b) Reachable set computation for infinitely many inputs.: The bifurcation seen in Figure 1(b) and in Figure 4(b) occurs when the stimulation current increases from $10\mu\text{A}/\text{cm}^2$ to $11\mu\text{A}/\text{cm}^2$. When the current reaches $11\mu\text{A}/\text{cm}^2$, an AP is fired, while for a current of $10\mu\text{A}/\text{cm}^2$, only a small bump is observed.

We conducted two sets of reachability analysis on the LHA model: one with the input current ranging within interval $[0, 10]\mu\text{A}/\text{cm}^2$ and the other with the input current ranging within interval $[0, 11]\mu\text{A}/\text{cm}^2$. The output of d/dt's reachability algorithm is a rendering of the shape of the reachable set in each mode. For a one-dimensional system such as LHA1, the reachable set is planar (taking consideration of time axis).

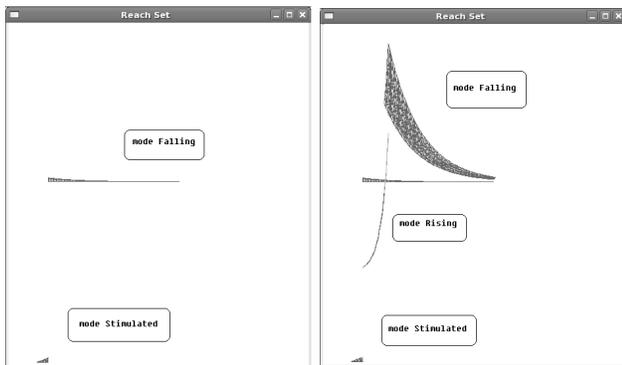


Fig. 5. (a) Reachable set for $I_{st} \in [0, 10]$. (b) Reachable set for $I_{st} \in [0, 11]$

Figure 5(a) shows the reachable set for a stimulus current between $0\mu\text{A}/\text{cm}^2$ and $10\mu\text{A}/\text{cm}^2$. Only modes `Falling` and `stimulated` are reached during the computation. Figure 5(b)

shows the reachable set for a stimulation current between $0\mu\text{A}/\text{cm}^2$ and $11\mu\text{A}/\text{cm}^2$. In this case, we see that all three modes are reached. In particular, bifurcation is observed in mode `Falling`: the reachable set for this mode is composed of an upper part, reached by switching from mode `Rising`, when an AP occurs, and a lower part, reached by returning from mode `Stimulated`, when an AP fails to occur.

V. CONCLUSIONS

We have presented a novel approach to investigating key behavioral properties of complex biological systems by first using automated techniques to learn a simplified *Linear Hybrid Automaton* model of the system under investigation, and then carrying out automatic reachability analysis on the resulting model. The merit in our approach resides in the fact that by performing symbolic reachability analysis on a Linear Hybrid Automaton, one can explore the system's reaction to a large, potentially infinite set of inputs. Any interesting behaviors revealed by such an analysis can then be validated via simulation of the original nonlinear system on the inputs to the LHA that gave rise to these behaviors. Ideally, we would like to have a soundness result that states that if no undesirable behaviors are observed in the abstracted LHA model, then none are present in the original nonlinear system. The derivation of such a result is work in progress and will likely follow lines of reason similar to those in [4], where we showed how to automatically learning an HA that approximates EC behavior up to a required error margin.

REFERENCES

- [1] R. Alur, C. Coucoubetis, N. Halbwachs, T. Henzinger, P. Ho, X. Nicolin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Sciences*, vol. 138, pp. 3–34, 1995.
- [2] N. Lynch, R. Segala, and F. Vaandrager, "Hybrid I/O automata," *Information and Computation*, vol. 185, no. 1, pp. 103–157, 2003.
- [3] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane currents and its application to conduction and excitation in nerve," *J Physiol*, vol. 117, pp. 500–544, 1952.
- [4] R. Grosu, S. Mitra, P. Ye, E. Entcheva, I. Ramakrishnan, and S. Smolka., "Learning cycle-linear hybrid automata for excitable cells," in *Proc. of HSCC'07, the 10th International Conference on Hybrid Systems: Computation and Control*, ser. LNCS, vol. 4416. Pisa, Italy: Springer Verlag, April 2007, pp. 245–258.
- [5] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *Hybrid Systems: Control and Computation*, ser. LNCS. Springer, 2003.
- [6] J. G. Dumas and A. Rondepierre, "Modeling the electrical activity of a neuron by a continuous and piecewise affine hybrid system," in *Hybrid Systems: Computation and Control*, 2003, pp. 156–171.
- [7] T. A. Henzinger, "The theory of hybrid automata," in *Proceedings of the 11th IEEE Symposium on Logic in Computer Science*, 1996, pp. 278–293.
- [8] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete abstractions of hybrid systems," *IEEE*, vol. 89, pp. 971–984, 2000.
- [9] T. Dang, "Verification and synthesis of hybrid systems," Ph.D. dissertation, INPG, 2000.
- [10] P. Ye, E. Entcheva, R. Grosu, and S. Smolka, "Efficient modeling of excitable cells using hybrid automata," in *CMSB'05, Computational Methods in Systems Biology Workshop*, Edinburgh, UK, April 2005.